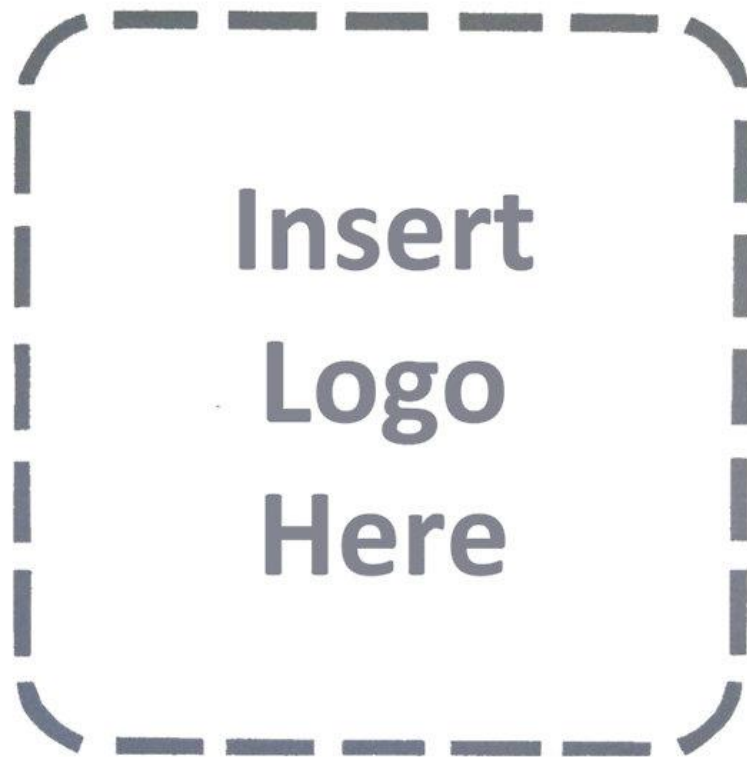


10 FEVRIER 2019



PFE : THE INDEPENDENT
SUPERVISOR
DII -2018/2019

BRANDON SIMON-VERMOT
ENCADRANT : M. AMEUR SOUKHAL

Commentaire	Date de modification
Création du document	06/02/2019
Ajout des liens annexes	06/02/2019
Ajout de l'introduction, rappel sur le projet, livrables	06/02/2019
Ajout conclusion, difficultés rencontrées	06/02/2019
Ajout des remerciements	06/02/2019
Ajout des résultats frontend	06/02/2019
Ajout base méthodologique, évolutions, gestion de projet, conditions de fonctionnement	07/02/2019
Ajout réalisation API REST, glossaire	08/02/2019
Ajout SWOT, TOWS, MOSCOW, bibliographie, bilan financier	08/02/2019

// WIP

Rajouter plus de liens biblio / glossaire

Table des matières

Remerciements	4
Introduction	4
Livrables.....	5
Cahier de spécification.....	5
Cahier d'analyse.....	5
Rapport.....	5
Manuel d'utilisateur	5
Manuel d'administrateur.....	5
Manuel de développeur.....	6
Manuel de mainteneur	6
Plannings	6
Vidéo.....	6
Poster.....	6
Rappel sur le projet.....	7
Contexte.....	7
Objectifs du PFE.....	7
Bases méthodologiques	8
Description du projet	9
Environnement du projet	9
Caractéristiques des utilisateurs	10
Fonctionnalités et structure générale du système	11
Contraintes de développement, d'exploitation et de maintenance	11
Description des interfaces externes du logiciel	12
Interfaces homme/machine.....	12
Interface logiciel/logiciel.....	17
Conditions de fonctionnement.....	19
Performances.....	19
Capacités.....	19
Contrôlabilité.....	19
Sécurité	19
Gestion de projet	20
Choix de la méthode de développement	20
Matrice SWOT	20
Matrice TOWS	21
Matrice MOSCOW	21

Gantt	22
Gantt prévisionnel	22
Gantt réel.....	22
GitKraken.....	24
GitKraken Glo	25
Évolutions.....	28
Difficultés	28
Bilan Financier.....	29
Conclusion	29
Annexes.....	30
Lien vers le cahier d'analyse	30
Lien vers le cahier de spécification	30
Lien vers le cahier des charges	30
Lien de la vidéo de démonstration	30
Lien vers le projet GitHub	30
Glossaire	31
Bibliographie.....	32

Remerciements

Je tiens à remercier M. Ameer SOUKHAL qui sans sa confiance, ce projet n'aurait pas eu lieu dans ces conditions. Ainsi j'ai pu réaliser mon projet personnel grâce à une validation suite à son accord pour encadrer le projet.

Je remercie aussi tous les membres de la team « The Independent Gamers », qui ont su m'apporter une motivation et une confiance tout au long de ce projet, mais aussi leur fidélité depuis plusieurs années et ce depuis la création de cette équipe.

Je n'oublie pas de remercier, [Maximilian Schwarzmüller](#) qui est un développeur Web professionnel et instructeur qui donne un cours complet sur la pile MEAN, ce qui m'a beaucoup aidé dans l'apprentissage et la prise en main de ces technologies.

Un dernier remerciement va tout droit à ma conjointe, qui a su comprendre le besoin d'avoir du temps pour le projet mais aussi pour son temps passé à la réalisation d'un logo qui accompagnera le projet.

Introduction

Dans le cadre de notre formation en école d'ingénieur, à Polytech'Tours, nous devons réaliser un projet de fin d'étude. Pour cela plusieurs sujets nous sont proposés, projets proposés par les différents professeurs de l'école ou proposé par une entreprise. Cependant il est aussi possible de proposer un projet personnel, ce qui est mon cas.

Pour proposer un projet, il faut au préalable fournir un cahier des charges et que le sujet soit étudié puis validé par un jury, tout cela se passe avant le début d'année scolaire. Ainsi le sujet est évalué afin de déterminer s'il répond aux critères attendus.

Livrables

Les livrables du projet sont orientés pour chaque type d'utilisateur. Ils permettent à chaque acteur ou futurs développeurs de prendre en main la partie front-end, avec l'application Web et la partie back-end, avec l'API REST. Dans cette partie ces livrables seront détaillés afin de comprendre leurs rôles et leurs enjeux.

Cahier de spécification

Le cahier de spécification liste précisément ce qui sera fait au long du projet. Ce document est destiné aux personnes souhaitant comprendre les besoins, ainsi que les exigences du client. Les contraintes liées au projet font partie de ce document.

Cahier d'analyse

Le cahier d'analyse est un livrable qui explique en détail les choix techniques et solutions mises en place pour la réalisation du projet. On retrouve explicitement les choix et les études des différentes parties du projet. Ainsi ce document est complémentaire à l'analyse initiale effectuée lors de la phase de spécifications.

Rapport

Le rapport est le document qui fait le lien entre les différents livrables. Il permet de faire une synthèse sur le déroulement global du projet.

Manuel d'utilisateur

Le manuel d'utilisateur est un manuel d'utilisation de l'application. Il permet la prise en main étape par étape de l'application web.

Manuel d'administrateur

Ce manuel complète le manuel d'utilisateur, avec un point de vue d'administrateur, ainsi il complète la prise en main et décrit les interactions avec le parc d'applications.

Manuel de développeur

Si une personne souhaite reprendre le projet et ainsi développer de nouvelles fonctionnalités ou améliorer ce qui déjà présent, grâce à ce manuel, il sera possible de reprendre le projet dans sa globalité.

Manuel de mainteneur

Dans le but de déployer le projet et ces éventuelles mises à jour, des procédures sont indiqués dans ce manuel, afin d'assurer le bon déploiement de l'application.

Plannings

Deux plannings seront présentés, le planning initial et le planning réel. Ainsi il sera possible de comparer les deux plannings et voir le vrai déroulement du projet, cette comparaison, présent dans ce rapport, permettra d'expliquer les grandes différences et leurs raisons.

Vidéo

La vidéo résume le projet avec quelques explications sur le fonctionnement de l'application. Elle servira à montrer comment fonctionne le projet.

Poster

Le poster est une affiche qui décrit le projet et qui sera potentiellement affichée dans les bâtiments de Polytech'Tours afin d'informer les nouveaux étudiants et les visiteurs des projets réalisés auparavant.

Rappel sur le projet

Contexte

Dans le cadre de notre formation en école d'ingénieurs, nous devons réaliser un projet de fin d'études durant notre 5e année. Ce projet débute mi-Septembre 2018 correspondant à notre début de période école pour cette dernière année, pour finir à la fin de notre période école, pendant le mois de Février 2019. Étant donné que le projet se déroule en même temps que la période école, il n'est pas possible de travailler à plein temps sur le projet, une demi-journée sera donc consacrée au minimum chaque semaine.

The Independent Gamers est une équipe de joueurs amateurs, cofondé en 2013 par Brandon SIMON-VERMOT, cette équipe partage l'idée de fournir une ambiance divertissante en proposant diverses activités et la possibilité de rencontrer des joueurs réguliers sur une panoplie de jeux divers. The Independent Supervisor est le fruit d'une idée venant du chef de l'équipe de joueurs amateur qui possède un serveur dédié, sur lequel plusieurs serveurs de jeux sont installés.

- Brandon SIMON-VERMOT : Maitre d'œuvre
- Aneur SOUKHAL : Encadrant de projet & maitre d'ouvrage

Pour plus d'informations de manière globale, la lecture du cahier de spécification peut s'avérer utile à la bonne compréhension du projet.

Objectifs du PFE

Le besoin de ce projet est de superviser les serveurs de jeux et offrir la possibilité de les mettre à jour automatiquement, de les maintenir. Pour ce faire, des lancements, des arrêts d'application (via des commandes enregistrées associées à une application par exemple) seront exécutés automatiquement pour remplacer la maintenance manuelle, ce qui est fait actuellement. Ce projet part de zéro, aucun site Web et les outils liés (serveur Web, BDD, ...) ne sont déjà installés sur le serveur dédié, seuls les serveurs de jeux sont actuellement présents. Cette application web permettra donc de pouvoir exécuter une application ou l'arrêter grâce aux informations enregistrées par les administrateurs de l'application Web. Notamment les informations autour de chaque serveur de jeu (emplacement du dossier du serveur, commande de lancement, commande de sauvegarde), qui ici dans le cadre de ce projet seront les applications du parc soumis à la supervision.

Bases méthodologiques

Outils :

- VMware
- TeamViewer
- Mozilla Firefox / Google Chrome
- Visual Studio Code
- Terminal Shell
- GitKraken
- GitKraken Glo (équivalent Trello)
- Draw.io (UML)

Langages :

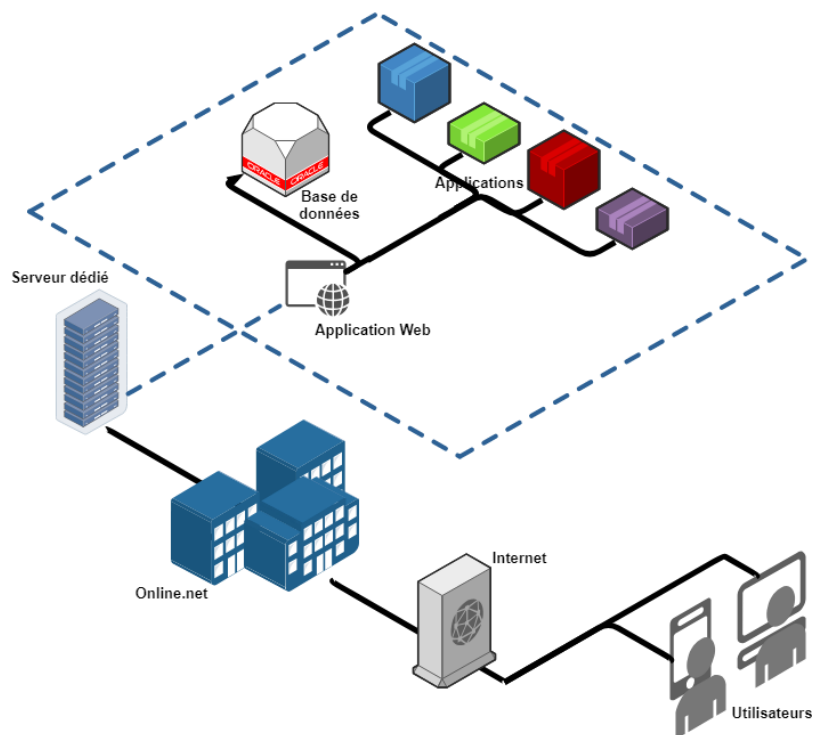
- TypeScript
- HTML
- CSS
- JavaScript
- Shell

Librairies :

- Express
- Mongoose
- Angular
- Angular Material
- BcryptJS
- JsonWebToken
- Diskspace
- Multer
- Ng2 Charts
- Respawn
- Nodemon

Description du projet

Environnement du projet



Ci-dessus, on peut retrouver l'environnement du projet disponible à la fin de ce projet, dont le travail au cours de ce dernier a été concentré sur l'application Web et l'API REST, mais aussi la base de données NoSQL qui sont tous présent sur le serveur dédié.

D'autres fonctionnalités, notamment la communication (via mail, réseaux sociaux, etc.) aurait dû apparaître, un rappel sur ce point est réalisé plus bas, dans les améliorations envisagées, dans ce rapport.

Caractéristiques des utilisateurs

- Utilisateur

Un utilisateur pourra consulter le parc d'applications et connaître leur statut, ainsi il pourra reconnaître les applications qui sont disponibles. Il est possible pour un utilisateur de poster une suggestion ou un commentaire afin de régler des soucis de bug que ça soit sur l'application Web ou sur une des applications, mais cela peut être aussi une demande de modification de configuration d'une application.

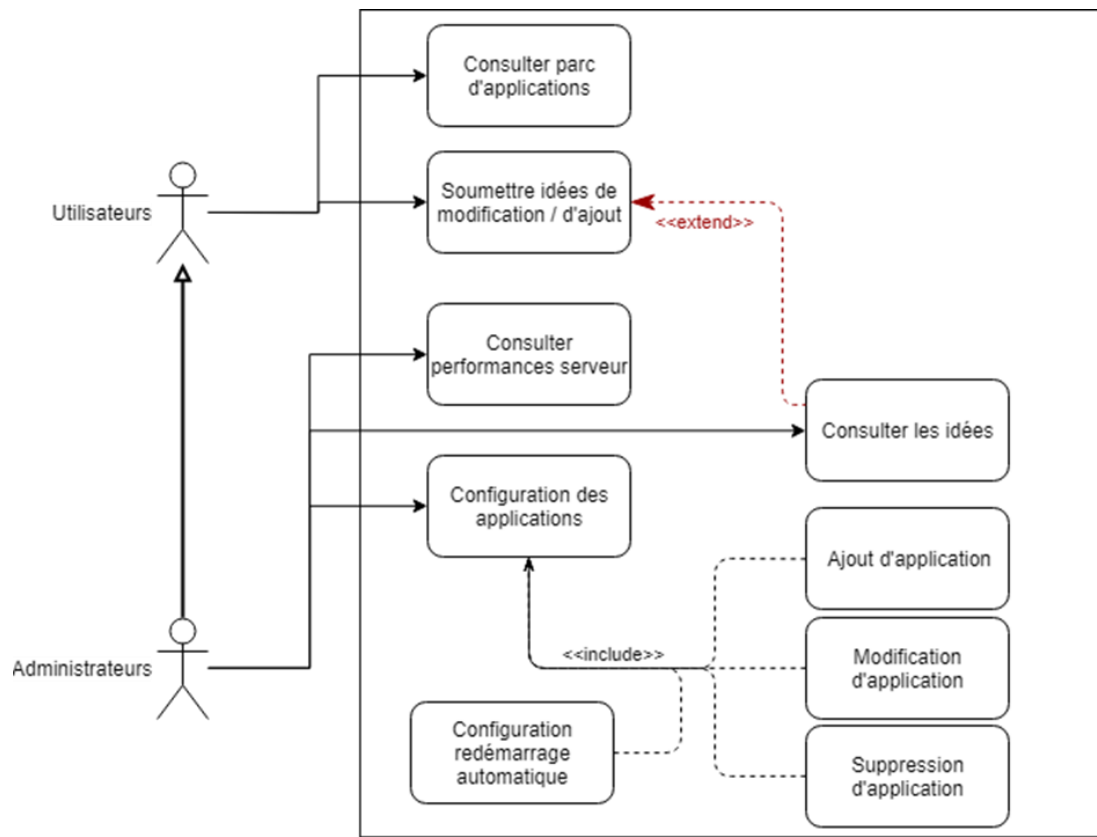
- Administrateur

L'administrateur pourra réaliser les mêmes actions qu'un utilisateur mais pourra aussi consulter les performances du serveur dédié, éteindre ou lancer une application, en mode simple ou auto-run, ce qui permettra que l'application redémarre en cas de crash. D'autres actions autour du parc sont disponibles, il est notamment possible de mettre à jour une application éteinte.

Cette personne pourra n'avoir aucune connaissance en programmation. Néanmoins, il est nécessaire qu'elle puisse savoir accéder à une page internet et savoir naviguer sur l'application afin d'y réaliser différentes actions.

L'interface Homme-Machine doit donc être facile à utiliser et à comprendre.

Fonctionnalités et structure générale du système



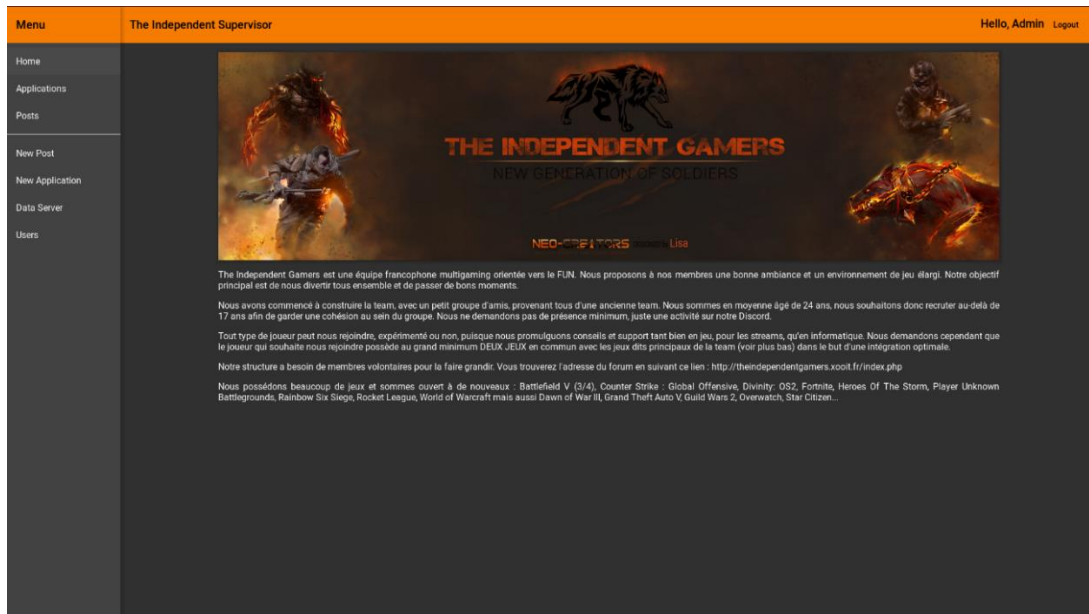
Contraintes de développement, d'exploitation et de maintenance

Le développement de l'application a été effectué afin de respecter une contrainte sur l'accessibilité de l'application Web. En effet celle-ci doit être accessible depuis la plupart des navigateurs, ainsi que sur mobile.

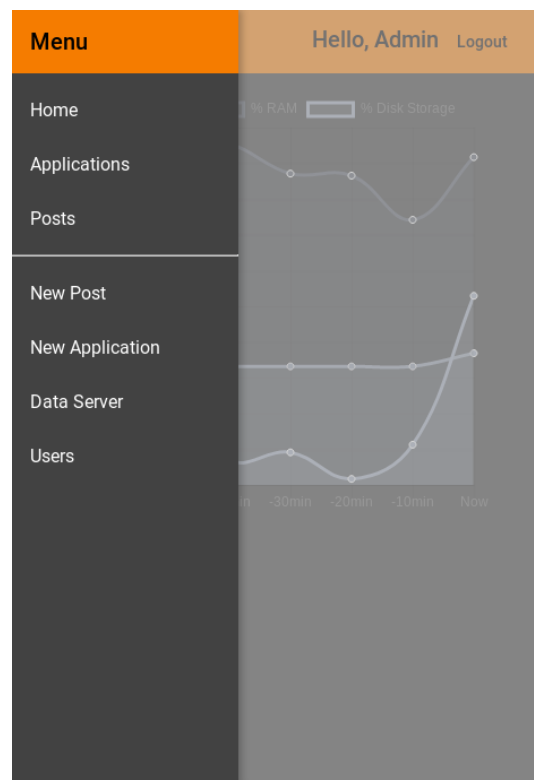
Pour le système d'exploitation, ce n'était pas imposé mais au vu du nombre d'applications tournant sous Linux, indiqué par le client, il est plus intéressant pour un serveur dédié de s'orienter vers ce dernier.

Description des interfaces externes du logiciel

Interfaces homme/machine



Ci-dessus, on peut retrouver le menu d'accueil de notre application depuis un navigateur web par la suite seules les interfaces vues depuis un mobile seront présentées, elles sont toutes développées en HTML/CSS qui sont rendues dynamiques grâce à du TypeScript. L'équivalent de l'interface ci-dessus ressemble à ceci :



The image displays two mobile application screens side-by-side, both featuring an orange header bar with a hamburger menu icon, the text 'TIS', and links for 'Login' and 'Sign Up'.

The left screen is the 'Sign Up' interface. It contains a dark gray form box with three input fields: 'E-Mail *', 'Username *', and 'Password *'. A yellow 'Sign Up' button is positioned at the bottom of the form.

The right screen is the 'Login' interface. It contains a dark gray form box with two input fields: 'E-Mail *' and 'Password *'. A yellow 'Login' button is positioned below the 'Password *' field.

Afin de créer un utilisateur, il est possible de le faire depuis l'interface se trouvant à gauche, cependant cette interface est temporaire, pour la suite du projet, seuls les administrateurs pourront donner accès à l'application Web, pour donner les droits administrateurs, tout se fait pour le moment via la base de données en modifiant une information via une communication direct.

Pour se connecter, l'interface est presque similaire, seul l'information du nom d'utilisateur n'est pas apparente, car l'adresse mail et le mot de passe suffit pour reconnaître et vérifier les informations, par la suite l'adresse mail pourra être remplacer par le nom d'utilisateur et le mot de passe, ou il sera possible de se connecter via l'une des deux informations.

La connexion d'un utilisateur expire au bout d'une heure par défaut, cette information est stockée en locale via les données du navigateur web du client.

TIS

Hello, Admin

Logout

Post Title

Pick Image

Post Content

Save Post

TIS

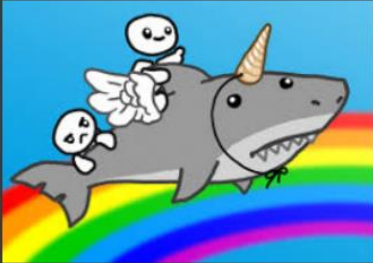
Hello, Admin

Logout

test12

coucou

hey



Super content !

EDITDELETE

Items per page: 5

1 - 3 of 3

<>

Une fois connecté en tant qu'utilisateur, il est possible à ce dernier d'ajouter un message, afin d'informer d'un éventuel problème, ou d'une demande de modification d'une application, voire d'en ajouter, plusieurs utilisations sont possibles. L'ajout de cette suggestion s'accompagne d'un titre, d'un contenu et d'une image. Seul le créateur et les administrateurs peuvent éditer ce message, ou le supprimer.

The screenshot shows the 'Add Application' form in the TIS admin interface. The header is orange with a hamburger menu, 'TIS', 'Hello, Admin', and a 'Logout' link. The form is a dark gray box with the following elements: a text input for 'Application Title', a 'Pick Image' button, a text area for 'Application Content', and a yellow 'Save Application' button at the bottom right.

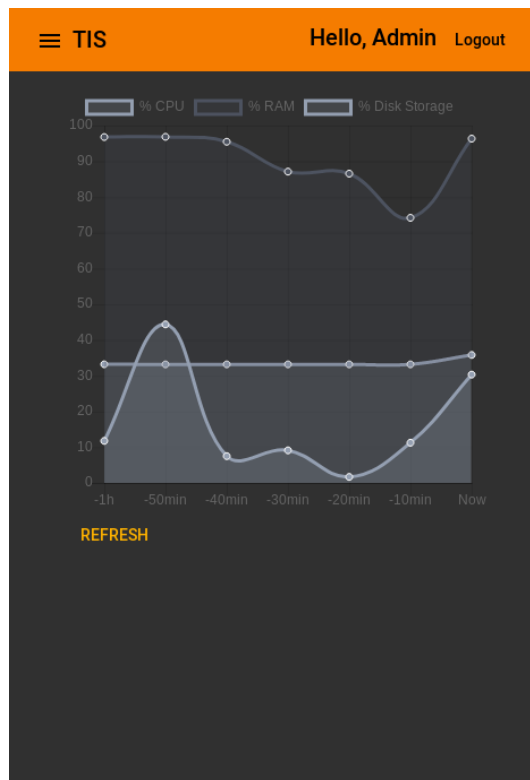
The screenshot shows the application list in the TIS admin interface. The header is orange with a hamburger menu, 'TIS', 'Hello, Admin', and a 'Logout' link. The main content area displays a card for 'TrackMania² Stadium' (TM2). The card features a game image, a description in French, and a row of five icons: a green play button, a green refresh button, a yellow circular arrow, a pencil, and a red trash can. At the bottom, there is a pagination bar showing 'Items per page: 5' and '1 - 1 of 1'.

Dans le cas de l'ajout d'une application, seul un administrateur peut réaliser cette action, une fois sauvegarder, plusieurs fichiers sont générés afin qu'une personne ayant accès au serveur dédié puisse modifier ces fichiers. Ces derniers sont des scripts Shell, qui vont permettre de lancer l'application, la mettre à jour ou encore éteindre cette dernière.

Une fois l'application et les scripts correspondant, l'administrateur peut accéder aux différentes actions depuis la liste des applications. Il est donc possible de lancer en mode simple, ce qui permet de lancer une seule fois l'application et si elle s'éteint pour une raison X ou Y, rien ne relancera l'application. Le mode auto-run, permet de contrer ce genre de soucis, il permet de relancer l'application dès que cela arrive. Cependant par défaut, une limitation de trois redémarrages est affectée pour chaque application lancée via ce mode, avant qu'elle s'arrête définitivement.

Si une application est lancée, ces actions ne sont plus disponibles, elles sont remplacées par l'action permettant de relancer l'application et celle permettant de couper l'application.

Il est possible de mettre à jour l'application depuis ce menu, côté software qui elle aussi n'est plus disponible une fois l'application lancée, mais aussi éditer les informations de l'application, tout comme la supprimée.



Une des fonctionnalités importantes, disponible seulement pour les administrateurs, leur permet de consulter un graphique indiquant sur une heure défilante, le taux d'utilisation du processeur, de la mémoire vive et celle du disque dur. À partir de ces informations, si une des données dépasse à l'instant présent, une certaine limite, fixé par défaut à 80%, cela coupe la dernière application lancée, afin de réduire la charge sur le serveur dédié.

Interface logiciel/logiciel

Un script Shell est lancé pour chaque action avec une application du parc, cela correspond à un lancement ou une mise à jour, il est possible aussi de configurer un script pour l'arrêt d'une application, afin qu'elle arrête plus proprement une application.

Une base de données NoSQL est présente, afin de stocker les différentes informations autour des suggestions postés, des applications mais aussi les données d'utilisation autour du serveur dédié.

Afin d'accéder à ces données, une API REST a dû être mise en place, celle-ci développée en JavaScript dans le but que ça soit hébergé via NodeJS. Plusieurs routes ont dû être définies :

Méthode	Route	Commentaire
GET	/api/apps	Permet d'obtenir toutes les applications enregistrées
GET	/api/apps/:id	Permet d'obtenir les informations de l'application associé à l'id
POST	/api/apps	Permet de créer et d'enregistrer les informations d'une application
PUT	/api/apps/:id	Permet de mettre à jour les informations de l'application associé à l'id
PUT	/api/apps/:id/run	Permet de lancer le script pour démarrer l'application associé à l'id
PUT	/api/apps/:id/stop	Permet de lancer le script pour arrêter l'application associé à l'id
PUT	/api/apps/:id/update	Permet de lancer le script pour mettre à jour l'application associé à l'id
DELETE	/api/apps/:id	Permet de supprimer les informations associées à l'id

Méthode	Route	Commentaire
GET	/api/data-server	Permet d'obtenir les dernières informations des performances du serveur dédié

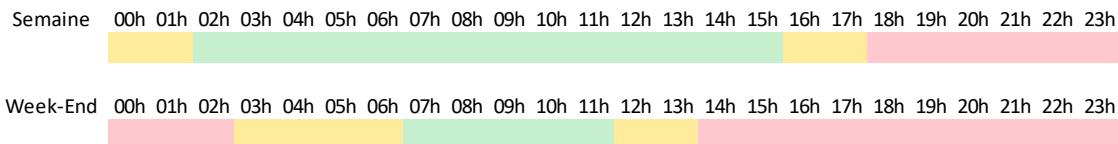
Méthode	Route	Commentaire
GET	/api/posts	Permet d'obtenir tous les suggestions enregistrées
GET	/api/posts/:id	Permet d'obtenir les informations de la suggestion associé à l'id
POST	/api/posts	Permet de créer et d'enregistrer les informations d'une suggestion
PUT	/api/posts/:id	Permet de mettre à jour les informations de l'application associé à l'id
DELETE	/api/posts/:id	Permet de supprimer les informations associées à l'id

Méthode	Route	Commentaire
GET	/api/users	Permet d'obtenir tous les utilisateurs enregistrées
GET	/api/users/:id	Permet d'obtenir les informations d'un utilisateur enregistrées
POST	/api/users/signup	Permet de créer un utilisateur
POST	/api/users/login	Permet de se connecter via la génération d'un JWT (Json Web Token)

Conditions de fonctionnement

Performances

- L'application Web ainsi que les applications du parc, qui sont lancées, devront fonctionner en continu durant une durée indéterminée.
- Les temps d'attentes ou d'exécution des actions ne devront pas gêner l'utilisation de l'application Web par les visiteurs authentifiés.
- L'utilisation de l'application peut être très fréquente par moments et nulles d'autres fois, on a pu déterminer un planning de fréquentation :



Verte : faible fréquentation, Jaune : fréquentation moyenne, Rouge : forte fréquentation

Capacités

- Les interactions avec l'utilisateur devront être fluides.
- L'application Web doit pouvoir gérer et stocker les données des suggestions et des applications du parc.

Contrôlabilité

Le taux d'utilisation du serveur dédié, est affiché en continu afin que les administrateurs qui ont accès au server puisse intervenir afin de vérifier qu'il n'y ait pas de soucis externes à l'application Web et les applications qui sont lancées.

Sécurité

Pour le bon fonctionnement de l'application, deux niveaux d'utilisateurs sont définis afin de limiter les accès de l'application aux utilisateurs qui ne sont pas administrateurs. Tel que les actions suivantes ne sont pas disponibles : ajout et modification d'une application, lancement d'une application, arrêter une application.

Gestion de projet

Afin de mener à bien ce projet, dans le temps imparti, il a été nécessaire de réfléchir à la méthode et aux outils de gestion de projet qu'il fallait utiliser. Une fois ce choix effectué, une découpe des tâches s'est révélée importante pour en déterminer un planning prévisionnel, tout en prenant en compte les différents jalons imposés pour chaque livrable.

Choix de la méthode de développement

Au vu de la structure des projets de fin d'études, notamment avec la rédaction d'un cahier de spécification et d'un cahier d'analyse, le cycle en V est la méthode la plus adéquate. De cette manière, plusieurs jalons peuvent en sortir afin de valider les différentes étapes de progression dans le développement du projet.

Matrice SWOT

<u>Forces</u>	<u>Faiblesses</u>
Compétences en informatique solides Projet intéressant qui peut venir répondre à un besoin commun	Langages de programmation non maîtrisés (TypeScript, JavaScript, ...)
<u>Opportunités</u>	<u>Menaces</u>
Se faire connaître Publier l'application Réaliser un projet qui peut être utile dans plusieurs cadres d'application	Mauvais choix de stratégie impliquant un retour en arrière dans la démarche

L'intérêt de ce projet est qu'il adaptable en fonction du besoin, dans le cadre de mon projet, les applications cibles sont des serveurs de jeux, ayant besoin d'être superviser et maintenable facilement. Cependant cela pourrait très bien correspondre à une entreprise qui souhaite avoir les mêmes fonctionnalités pour des applications internes, par exemple.

Dans le but d'avoir une application au goût du jour, il était intéressant d'avoir des langages fortement utilisés actuellement, cependant ce genre de choix peut montrer des surprises par la jeunesse d'un langage et ainsi causer des soucis de stratégie, pour contrer ce problème, une analyse de la faisabilité a pu être faite avant l'élaboration d'un cahier d'analyse.

Tout comme l'apprentissage d'un nouveau langage est dangereux, si on ne sent pas apte à sortir de sa zone de confort, dans mon cas l'apprentissage et la découverte de nouvelle technologie est ma motivation principale, ainsi depuis plusieurs années j'ai pu acquérir et consolider des compétences en informatique.

Matrice TOWS

Suite à la constitution de la matrice SWOT et trouver une solution pour supprimer les faiblesses et atténuer les menaces grâce aux opportunités et aux forces, nous pouvons déterminer la matrice TOWS :

	<u>Forces</u> N°1 : Besoin commun N°2 : Compétences en informatique	<u>Faiblesses</u> N°1 : Nouvelles notions
<u>Opportunités</u> N°1 : Faire connaître l'application N°2 : Découverte nouvelles notions	Présentation des résultats	Suivi d'un cours complet
<u>Menaces</u> N°1 : Mauvais choix stratégique	Application flexible	Analyse importante

Matrice MOSCOW

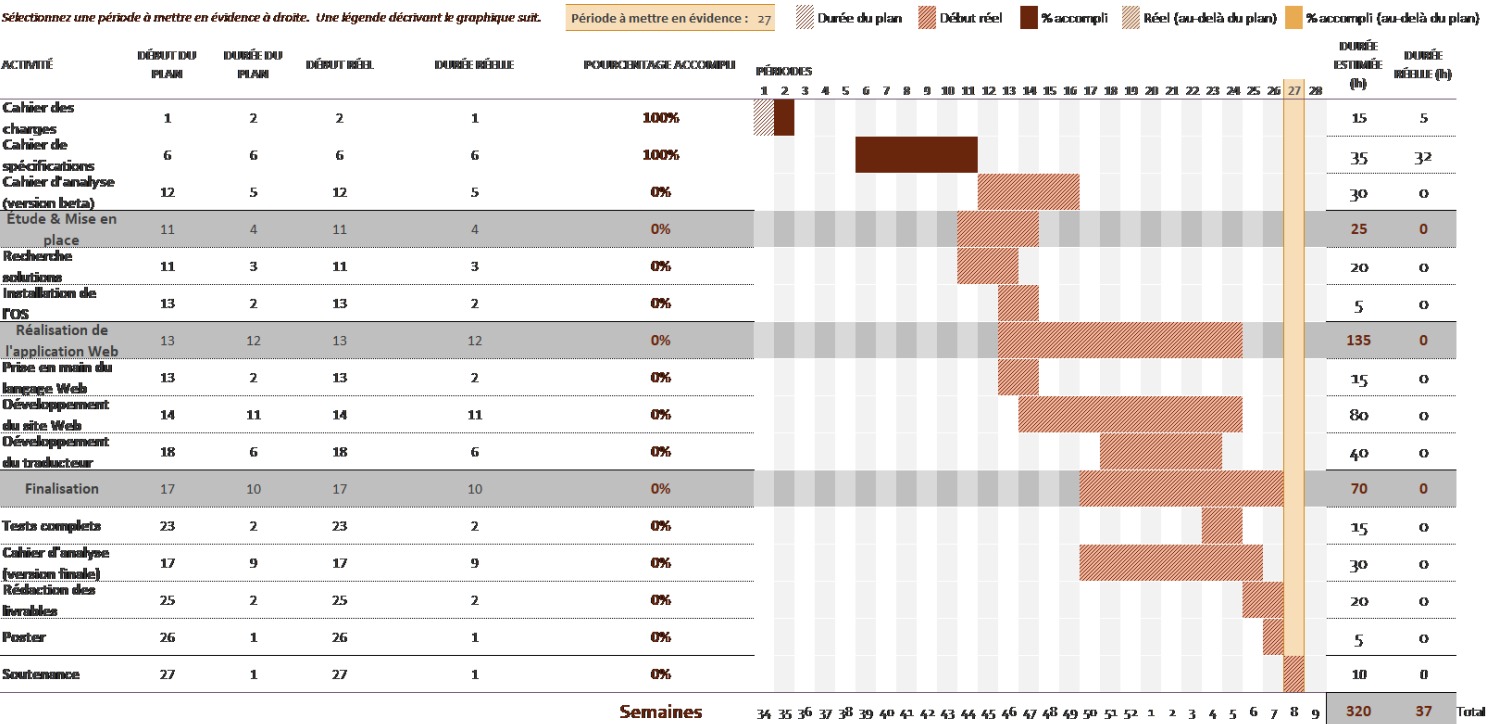
<u>Must</u> Démonstration de faisabilité	<u>Should</u> Réalisation de l'application Web et de l'API REST
<u>Could</u> Développement du traducteur	<u>Won't</u> Mise en production

Après la réalisation de la matrice SWOT, TOWS, une matrice MOSCOW a été réalisée par la suite pour rendre les réponses données aux faiblesses et aux menaces réalisables et cohérentes. Ainsi cette matrice a pu permettre de déterminer les différentes tâches accompagnées de leur niveau d'importance, qui à partir de ces tâches le planning prévisionnel a pu être réalisé.

Gantt

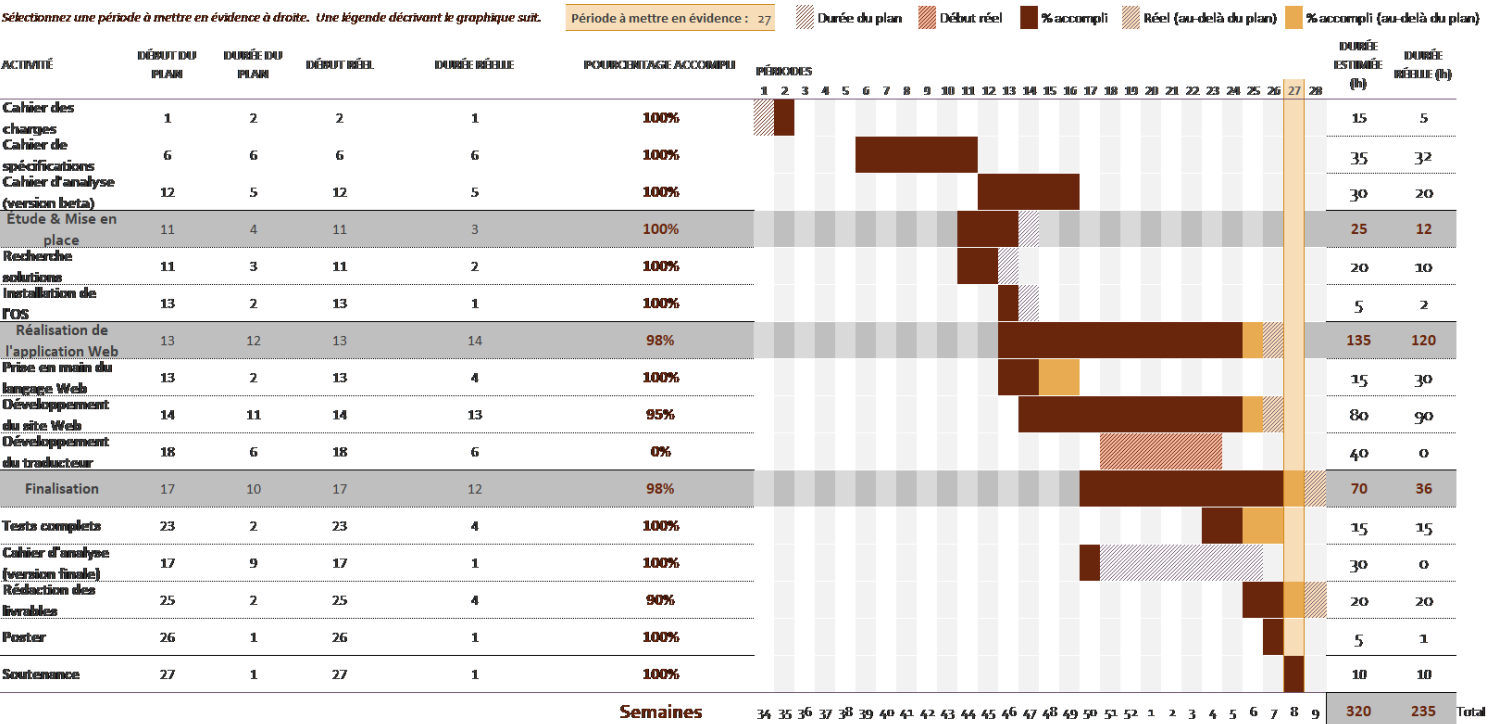
Gantt prévisionnel

Sélectionnez une période à mettre en évidence à droite. Une légende décrivant le graphique suit.



Gantt réel

Sélectionnez une période à mettre en évidence à droite. Une légende décrivant le graphique suit.



Comme on peut le remarquer de l'avance a été prise sur la recherche de solution et l'installation du système d'exploitation mais que cette avance n'a pas pour autant empêcher d'avoir du retard sur le développement de l'application Web.

Ce retard s'explique sur le fait que la technologie était nouvelle pour moi et de ma faible expérience dans le développement Web d'une application concrète qui va plus loin que les simples exercices qu'on peut réaliser pour apprendre à développer son premier site, ce qui rajoute de la complexité à prendre en compte sur le temps de compréhension et de la prise en main.

Le développement du traducteur a été omis, suite au manque de temps et de l'augmentation de la charge de travail via les différents contrôles continus qui sont nouveaux dans le cursus de cette dernière année d'école d'ingénieur. Dans la suite logique de ce retard, le temps passés pour les tests a augmenté proportionnellement.

Il n'y a pas eu de modification sur le cahier d'analyse, ainsi j'ai pu me concentrer sur l'application et gagner du temps de développement. Les deux dernières semaines, le travail est concentré sur la rédaction des différents livrables, cependant le cahier des livrables et les différents manuels seront rendus après la soutenance et donc la date de rendue de ce rapport.

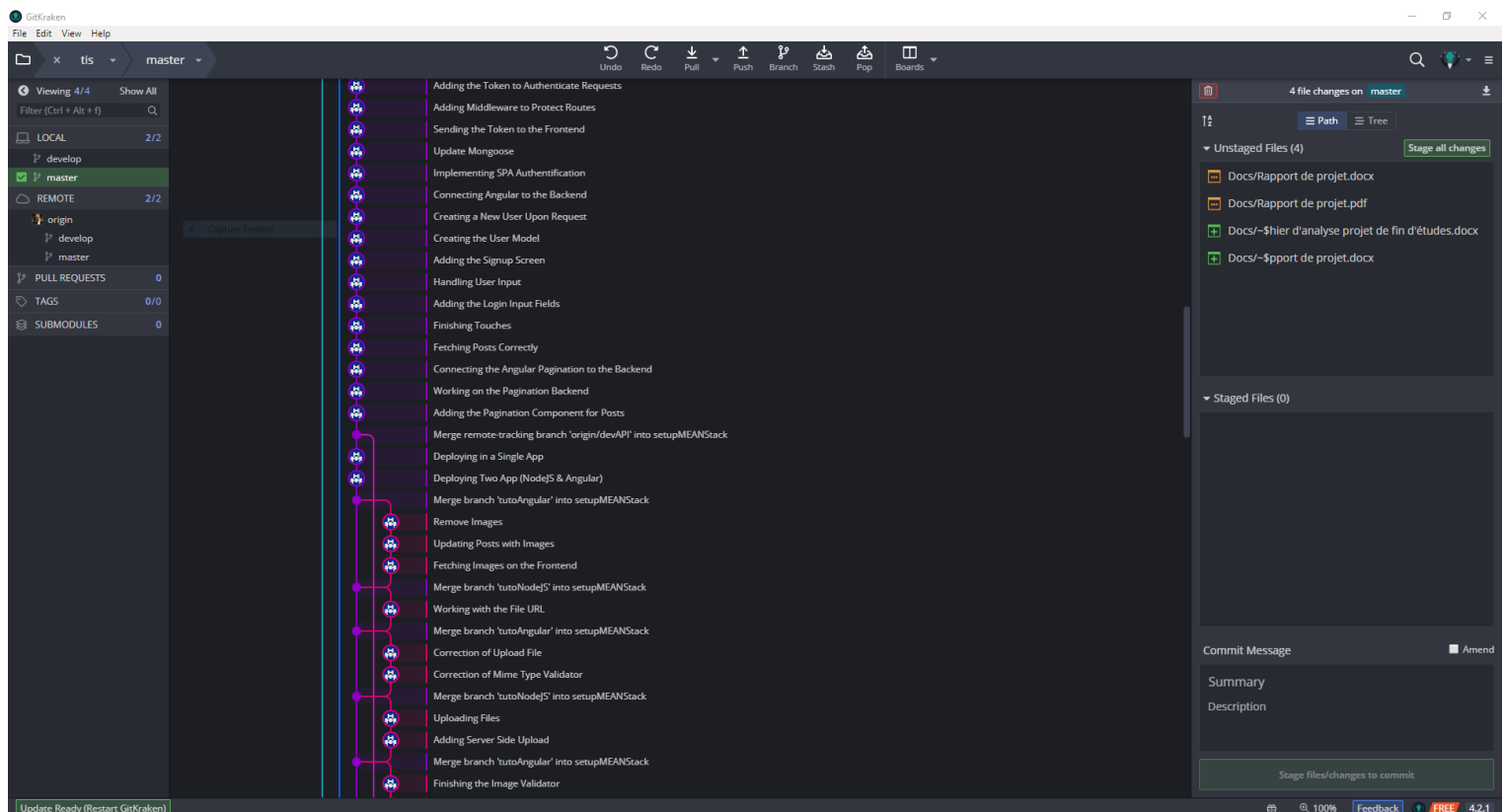
GitKraken



GitKraken est un client graphique pour le logiciel de versionnalisation Git. Après une rapide installation, et une configuration très assistée, le logiciel nous propose une liste de tous les dépôts Git présent sur le poste de travail, ou via son compte GitHub, et autres sites.

De là, il est possible de retrouver toutes les fonctionnalités qu'on peut attendre d'un client Git : navigation et aperçu de toutes les branches et commits, ajouts et modifications de remotes, diff, ... Le logiciel est très bien fait et assez clair pour que les fonctionnalités dont nous avons besoin soit facilement accessible quand nous en avons besoin.

GitKraken est un bon outil pour aider à manipuler git tous les jours si on n'est pas habitué à la ligne de commande. Les fonctionnalités ont l'air toutes présentes, graphiquement le logiciel est agréable, et surtout son installation et son utilisation sont très simples.



GitKraken Glo



L'interface utilisateur Glo a été conçue très intentionnellement avec le panneau de détails sur le côté droit de l'écran pour fournir un moyen non intrusif d'afficher plus d'informations. Il se met à jour instantanément lorsque vous cliquez sur une autre carte. Contrairement à Trello qui ouvre un modal et verrouille votre écran lorsque vous cliquez pour voir une carte. Ce n'est peut-être pas un défaut fatal, mais c'est frustrant dans les moments où vous voulez simplement parcourir un tas de cartes, vous avez cliqué sur la mauvaise carte ou vous voulez simplement vérifier une autre carte avant de faire une modification.

Bien qu'il soit important d'avoir tout ce dont vous avez besoin dans une seule vue, cela signifie aussi que vous devez être capable de filtrer rapidement ce que vous n'avez pas besoin de voir !

Avec la recherche de Trello, vous tapez d'abord votre chaîne de recherche, puis vous appuyez sur Entrée pour consulter les résultats. Si vous voulez utiliser des filtres à cartes, vous les trouverez enterrés dans le menu. Alternativement dans Glo, vous commencez simplement à taper et les résultats de votre recherche sont mis à jour avec chaque caractère supplémentaire, et les filtres de cartes fonctionnent juste là dans votre barre de recherche ! Avec de telles fonctionnalités, vous trouverez sans aucun doute ce que vous cherchez plus rapidement dans Glo que dans Trello.

Lorsqu'il s'agit de gérer vos tâches, vous voudrez savoir ce qui vous est assigné spécifiquement. Dans Trello, lorsque vous cliquez sur l'icône d'un utilisateur, il ouvre un modal pour visualiser l'activité de cet utilisateur et quelques autres choses (voir capture d'écran ci-dessous). Alternativement, lorsque vous cliquez sur l'icône d'un utilisateur dans Glo, il filtrera votre forum pour n'afficher que les cartes assignées à l'utilisateur sélectionné. Et parce que les cartes peuvent être assignées à plusieurs personnes, vous pouvez cliquer sur plusieurs nœuds pour voir les éléments assignés à un duo ou une équipe.

La plupart des développeurs sont très habitués à utiliser un calendrier pour voir quelles tâches à venir sont programmées pour le jour, la semaine et ainsi de suite ; c'est très utile pour planifier votre temps. Si vous voulez une vue du calendrier dans Trello, vous devrez d'abord appliquer le Power-Up Calendrier à votre compte. Si vous utilisez déjà votre seul Power-Up gratuit, vous serez invité à passer à Trello Business Class : un plan payant.

Dans Glo, il y a une vue de calendrier libre et intégrée. Cliquez simplement sur l'icône du calendrier pour voir les tâches avec les dates d'échéance à venir. Vous verrez les étiquettes en un coup d'œil et vous pouvez cliquer sur les cartes pour voir plus d'informations. Dans Glo, par défaut, le panneau de gauche affiche une liste de cartes sans date d'échéance, ce qui facilite l'identification des cartes qui ont besoin d'une date d'échéance. Il suffit de glisser-déposer pour les assigner quand ils sont dus.

Il y a 31 millions de développeurs dans la communauté GitHub qui collaborent sur le code ensemble, cela représente beaucoup de tâches. Les problèmes GitHub sont ce que beaucoup de ces développeurs utilisent pour suivre les tâches, les bugs et les fonctionnalités. Il est donc logique que votre traqueur de tâches soit capable d'extraire ces éléments pour vous. Trello ne supporte pas officiellement GitHub Issue Sync.

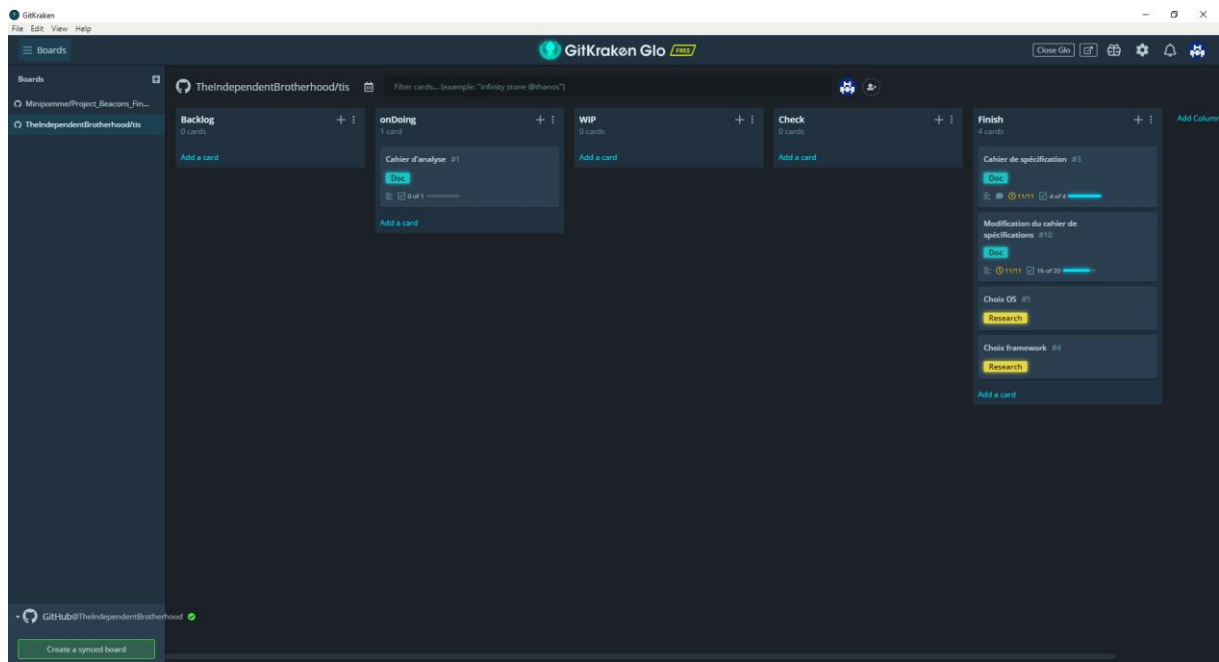
Alternativement, Glo fournit une intégration élégante, et tout comme la vue Calendrier, elle est incluse gratuitement. Lorsque vous synchronisez une carte Glo avec les problèmes GitHub, vous pouvez visualiser la progression des problèmes GitHub à travers votre flux de travail. Moins de changement de contexte est super important pour la productivité des développeurs.

Comme nous nous concentrons sur les développeurs, nous avons donné la priorité à l'accès facile à Glo dès le début. Glo peut être accédé directement dans le GitKraken Git Client, via un navigateur, ou en utilisant les plugins Visual Studio Code. C'est un changement de jeu pour la productivité car les développeurs peuvent accéder à leurs objets dans les outils qu'ils utilisent déjà ! Il vous suffit de faire une mise à jour de votre projet, puis de déplacer ou de mettre à jour votre carte Glo. Trello n'offre pas de plugins Visual Studio Code.

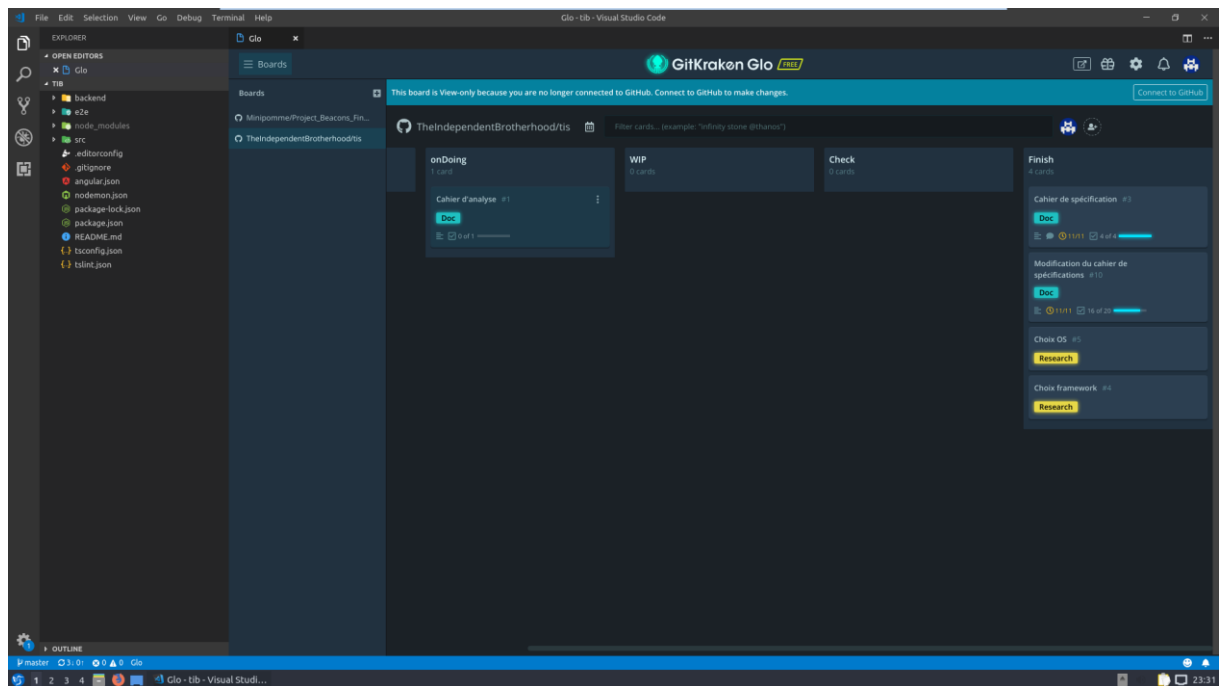
GitKraken vous permet de lier une carte Glo à votre dépôt actuel. Encore une fois, réduire le temps perdu à cliquer et à changer d'outil. D'un simple clic sur le bouton Glo du GitKraken Git Client, vous êtes toujours dirigé vers la planche Glo de votre projet. Trello n'offre pas de fonctionnalité comparable.

En fin de compte, tout dépend de votre façon de travailler et de l'outil qui répond à vos besoins. Nous avons vu une opportunité de faire le suivi des tâches différemment pour les développeurs parce que des outils comme Trello ne marchaient pas. Bien que la parité des fonctionnalités puisse changer, nous continuerons à utiliser notre expertise pour trouver des moyens de rendre le flux de travail des développeurs plus rapide et plus productif.

Vous devriez mettre les GitKraken Glo Boards à l'épreuve, et découvrir à quel point vous pouvez travailler plus efficacement ; c'est gratuit pour essayer toutes les fonctionnalités !

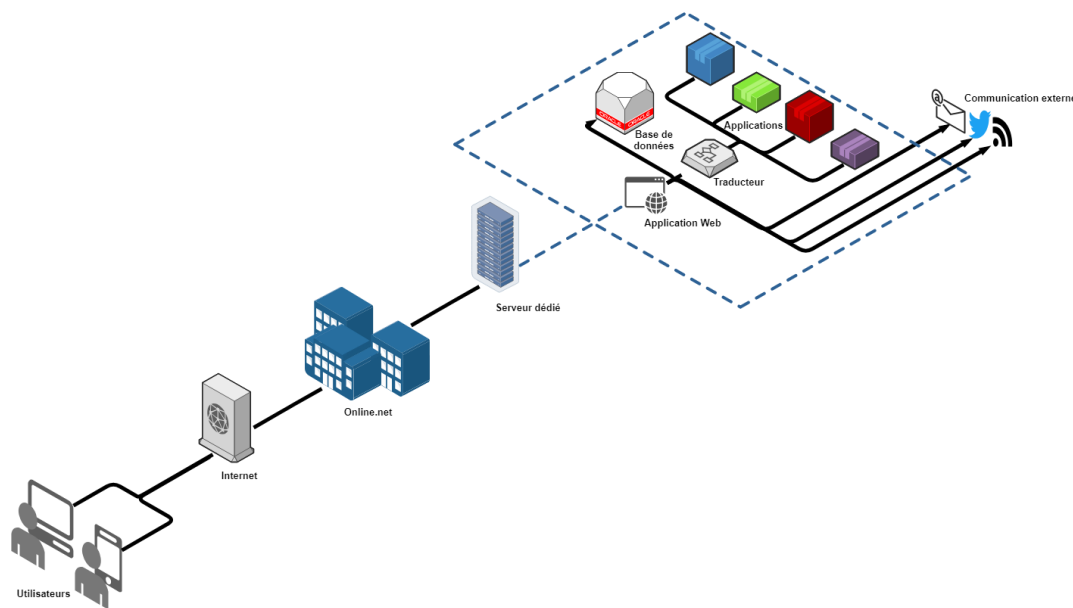


Vue de mon GitKraken Glo au début du projet



Vue de mon GitKraken Glo via le plugin pour Visual Studio Code, au début du projet

Évolutions



Le projet possède des fonctionnalités souhaitées qui sont encore à développer, notamment la mise à jour automatique avant chaque démarrage peu importe le mode (simple, auto-run, redémarrage).

Il serait intéressant de voir aussi sur une période plus précise ou non, les performances du serveur dédié et les différents taux d'utilisation. Puisqu'il est possible que l'administrateur ne soit pas présent pour consulter, quelle application ou à quelle période dans la journée, dans la semaine est la plus importante.

La communication externe est manquante pour signaler toute modification importante sur le parc d'application mais aussi pour les administrateurs, l'ajout de suggestion de la part d'un utilisateur.

Plusieurs améliorations sont possibles sur l'esthétique de l'application, mais ce n'est pas le point le plus important.

Difficultés

La principale difficulté rencontrée au long du projet est le fait de devoir réaliser plusieurs documents importants, ces documents ont une valeur proche de la réalisation technique qui se voit limitée par le temps dépensé à la réalisation de ces documents. À plusieurs reprises, une contrainte voit le jour pour réaliser certains objectifs techniques, ainsi cela impacte le résultat final de l'application.

L'apprentissage d'un nouveau langage et la prise en main d'une nouvelle technologie qui est Angular, m'a rendu plus difficile la réalisation technique pour la partie front-end. Cependant le suivi d'un cours de 28 heures m'a permis d'avoir une base tout au long de ce projet.

Bilan Financier

Une première description fut réalisée dans le cahier d'analyse, indiquant que seuls les coûts humains seront présents. Le serveur dédié est fourni par le client, ce qui n'engendre pas de coût supplémentaire. Le service DNS pour ainsi cacher l'adresse IP aux utilisateurs est un service gratuit qui par la suite pourra être changé si le client le souhaite, cependant cela reste à sa charge. Ce qui signifie que le coût du projet se résume à la prestation de main d'œuvre, avec un brut horaire à 8.17 € et un total d'heures estimée à 235 heures, cela fait un total de 1919.95 €.

Conclusion

Le projet est quasi fonctionnel sur tous les points, ainsi il répond à la plupart des objectifs souhaités par le client. Quelques réglages sont manquants pour l'esthétique de l'application ainsi que quelques règles pour la sélection d'une application à arrêter suite à une utilisation importante du serveur dédié... Cependant malgré le temps dépensé à la rédaction de plusieurs documents, l'application fournie réponds au fonctionnement de base, souhaité par le client.

Ainsi avec la réalisation de ce projet, j'ai pu découvrir et prendre en main, une nouvelle technologie. J'ai pu associer temps personnel et temps école pour développer un projet personnel que je souhaitais réaliser depuis plusieurs années.

Annexes

[Lien vers le cahier d'analyse](#)

[Lien vers le cahier de spécification](#)

[Lien vers le cahier des charges](#)

[Lien de la vidéo de démonstration](#)

[Lien vers le projet GitHub](#)

Glossaire

1. Json Web Token :

JWT est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. Elle s'effectue par l'algorithme HMAC ou RSA.

Un jeton se compose de trois parties :

- Un en-tête (header), utilisé pour décrire le jeton. Il s'agit d'un objet JSON.
- Une charge utile (payload) qui représente les informations embarquées dans le jeton. Il s'agit également d'un objet JSON.
- Une signature numérique.

Voici un exemple d'un JWT généré une fois connecté à l'application Web :

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFkbWluQGFKbWluLmZyIiwidXNlck1kIjoibW1ZGI5OGZiZTE1ZDAxNTU2ZjMxZTlhIiwiaXNBZG1pbSI6dHJ1ZSwiaWF0IjoxNTQ5NjQ2MzkxLCJleHAiOiE1NDk2NDk5OTF9.ynW9WWSOULm40rLOW1G0FsXDPpyjs7h8j0PUjDuTDg|
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "email": "admin@admin.fr",
  "userId": "5c5db98fbe15d01556f31e9a",
  "isAdmin": true,
  "iat": 1549646391,
  "exp": 1549649991
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```


Bibliographie

Cours pile MEAN : <https://www.udemy.com/angular-2-and-nodejs-the-practical-guide/>